CLAIMS

What is claimed is:

1. A method for data flow control of a plurality of execution nodes of an adaptive computing engine (ACE), the method comprising:

(a) associating a plurality of task parameters with a plurality of tasks within an execution node;

(b) identifying readiness of a plurality of task resources based on a status of the task parameters; and

(c) pacing allocation of the plurality of tasks to the execution node based on the readiness the plurality of task resources.

2. The method of claim 1 wherein the execution node includes a reconfigurable execution unit.

3. The method of claim 2 wherein the reconfigurable execution unit further comprises one or more finite state machines.

4. The method of claim 1 wherein the task parameters identify, by designation, an input port, an output port, a finite state machine, and a finite state machine instance.

5. The method of claim 4 wherein identifying a readiness step (b) further comprises the step of (b1) identifying a task as an executable task when the input port is available, the output port is available, and the finite state machine is idle.

6. The method of claim 1 further comprising the step of (d) aggregating executable tasks in a queue.

7. The method of claim 6 wherein allocation pacing step (c) further comprises the steps of (c1) reading a next executable task from the queue and (c2) generating a signal to start execution in the finite state machine associated with the next executable task.

8. The method of claim 7 further comprising the steps (e) of reconfiguring the finite state machine from one instance to another as necessary, reading data from the input port, (f) processing the data in the finite state machine, and (g) writing the data to the output port.

9. The method of claim 8 further comprising the steps of (h) generating a signal indicating completion of the execution in the finite state machine and (c) re-entering an idle state in the finite state machine.

10. The method of claim 4 wherein the designation comprises a number.

11. A system for flow control in processing nodes of an adaptive computing engine (ACE), the system comprising:

a reconfigurable execution unit; and

flow control logic coupled to the reconfigurable execution unit for associating tasks and task parameters, identifying readiness of task resources based on a status of the task

parameters, and pacing allocation of the tasks to the reconfigurable execution unit based on the readiness of task resources.

12. The system of claim 11 wherein the reconfigurable execution unit further comprises one or more finite state machines.

13. The system of claim 11 wherein the task parameters identify, by designation, an input port, an output port, a finite state machine, and a finite state machine instance.

14. The system of claim 13 wherein the designation comprises a number.

15. The system of claim 13 wherein the flow control logic further identifies a task as an executable task when the input port is available, the output port is available, and the finite state machine is idle.

16. The system of claim 12 further comprising a queue for aggregating executable tasks.

17. The system of claim 16 wherein the flow control logic reads a next executable task from the queue and generates a signal to start execution in the finite state machine associated with the next executable task.

18. The system of claim 13 wherein the finite state machine reconfigures from one instance to another, if necessary, reads data from the input port, processes the data, and writes the data to the output port.

19. The system of claim 18 wherein the finite state machine further generates a signal indicating completion of the execution and re-enters an idle state.

20. A system for flow control in processing nodes of an adaptive computing engine (ACE), the system comprising:

a plurality of finite state machines, each finite state machine for performing a task;

control logic for determining task parameter status for the task and identifying the task as executable; and

a task queue for storing executable tasks transferred by the control logic and issuing the executable tasks to the plurality of finite state machines.

21. The system of claim 20 wherein the plurality of finite state machines form an execution unit for a processing node within an adaptive computing engine.

22. The system of claim 20 wherein the control logic determines a status of an input port, an output port, a finite state machine idle state, and an instance of the finite state machine.

23. The system of claim 22 wherein the control logic identifies a task as executable when the input port and output port are available and the finite state machine is idle.